ProctorEx: An Automated Online Exam Proctoring System

Vinothini Kasinathan[1], Choi Ee Yan[1], Aida Mustapha[2], Vazeerudeen Abdul Hameed[1], Tham Hoong Ching[1], Vinesh Thiruchelvam[1]

[1] Asia Pacific University of Technology and Innovation, Taman Teknologi Malaysia, 57000 Kuala Lumpur, Wilayah Persekutuan Kuala Lumpur

vinothini@apu.edu.my, choieeyan@gmail.com, vazeer@staffemail.apu.edu.my, hctham@apu.edu.my, dr.vinesh@staffemail.apu.edu.my

[2] Faculty of Applied Sciences and Technology

Universiti Tun Hussein Onn Malaysia, KM1 Jalan Pagoh, 84600 Pagoh, Johor, Malaysia

aidam@uthm.edu.my

Article Info
Page Number: 876 – 889
Publication Issue:
Vol. 71 No. 3s2 (2022)

Abstract
Examination is a critical aspect of education, and online exams are no exceptions. One of the biggest challenges is the concern of cheating revolving around online examinations due to lack of proper invigilation as opposed to face-to-face exams. Online cheating is often undetected and difficult to prove, hence jeopardizing the academic integrity. To address this gap, this paper proposes for an automated online exam proctoring system that monitors students' browser activity and camera during an online exam. The proposed system will use intelligent face detection algorithms to detect any suspicious movement or activity and reports to an instructor in real time. This serves as a viable solution to reduce the chances of cheating going undetected and provides a means for instructors to monitor a large group of students at once during an online exam. The system is targeted towards any education institutions or individuals who wishes to assess their students' level of understanding or knowledge via online examinations.

Article History
Article Received: 28 April 2022
Revised: 15 May 2022
Accepted: 20 June 2022
Publication: 21 July 2022

Keywords: - Online learning, online exam monitoring, image processing, face detection, Covid19.

1. Introduction

One of the biggest challenges faced by educational institutions in e-learning is the concern of cheating surrounding online examination and assessments. The lack of invigilation during online exams has provided students the opportunity to cheat easily. During the COVID-19 pandemic where face-to-face exams were opted out, institutes were forced to devise different approaches to counter cheating during online exams [6]. Most of these approaches are rather ineffective or cumbersome. Meanwhile, many institutes across the world resort to postponing major exams or replacing them completely with assignments or open-book exams. It is clear to see that many of these institutes have low confidence in conducting exams online due to cheating concerns [5].

Online education is often deemed to be less credible than traditional learning before COVID-19 [5]. The shift to online learning driven by the pandemic has coerced people to experiment and explore with remote learning and its success could transform the future of education. Experts believed that online learning can very well be the future of education as technology becomes widely accessible [9]. However, cheating in online exams is still one of the leading concerns of these online institutions, resulting in most of them turning to open-book tests and exams instead. This inability to conduct online exams effectively is a limiting factor to the scalability of online learning in the education sector [1].

A comprehensive research was conducted by Dr. McCabe at the International Center for Academic Integrity throughout 12 years (2002-2015) on more than 70,000 graduate and undergraduate students across United States on academic dishonesty. The results were eye-opening, with 64 percent of students having admitted to cheating during tests and 58 percent admitted to plagiarism [4]. With the assistance of technology, students are given more opportunities to cheat today compared to students a generation ago. There are at least four ways in which students are cheating during online exams, those includes: accessing other websites, communicating with others, seeding computer with answers, and keeping exam answers nearby during the exam.

At present, human proctoring is the most common approach in overseeing an exam. In a traditional examination setting, one or more human proctors would invigilate a room of students. In online examinations, human proctors would monitor students visually and acoustically via student’s webcam. While having human proctors is effective in detecting and preventing cheating during exams, it is also labor-intensive and costly. Moreover, there is a limitation to how much a human eye can perceive at once. It is virtually impossible for human proctors to properly monitor many students during an online exam.

To alleviate the risk of cheating and plagiarism, this paper proposes for an automated web-based exam proctoring system that allows users to create and take exams that is automatically proctored using various detection algorithms. That way, instructors do not have to personally monitor every student as the system does the monitoring and sends real time alert when suspicious activity is detected. Through the implementation of the proposed system, instructors would be able to reduce instances of cheating going undetected during an exam.

The proposed system aims to provide instructors a simple yet effective means to supervise online exams through various methods like tab switching detection and detection of suspicious movements using Artificial Intelligence (AI) algorithms. For online institutions that often opt for open book exams due to online cheating, the system hopes to transform them by giving them the opportunity to consider giving students closed-book exams with the promise that the system would be able to capture and flag any suspicious action in real time.

The remaining of this paper proceeds as follows. Section 2 discusses works related to online examinations. Section 3 presents the underlying AI approaches in the proposed system, ProctorEx. Section 4 presents the ProctorEx prototype and discusses its functionalities. Finally, Section 5 concludes the paper.

2. Related Work

Proctoring generally refers to a person, known as the proctor, appointed to take charge in invigilating an exam to ensure that academic integrity guidelines are maintained during the exam. In conventional face-to-face exams, students are monitored by a human proctor within a classroom environment throughout the exam. In online proctored exams, a remote proctor observes and monitors exam takers over the internet through webcams to ensure no cheating is taking place undetected [8]. Online proctoring was first introduced by Kryterion (https://www.kryteriononline.com/).Since then, number of for-profit online proctoring services have skyrocketed, especially during COVID-19 when educational institutions all around the world were seeking for effective ways to prevent or catch online cheating.

[2] studied the likelihood of students cheating during un-proctored exams. During the experiment, an un proctored exam was held during which the Google search data was studied by the researchers simultaneously. The researchers found that exam related terms spiked through the roof the moment the test began, proving that students are very likely to cheat when knowing that they are not being proctored. Similar research also showed that students enrolling courses with proctored test scored on average 17 points lower than students enrolling same course with non-proctored test. As these studies suggested, online proctoring has been proven to be an effective method of preventing or catching cheating during online exam.

According to [7], commercial online proctoring solutions include the following approaches: live proctoring, recorded proctoring and automated proctoring. Live proctoring involves having a professionally trained human proctor to monitor an exam through webcam in real-time. Recorded proctoring involves a proctor reviewing video recordings of student taking exam. In automated proctoring, human proctors are replaced by proctoring software which leverages artificial intelligence or other algorithms to identify key events of possible fraud or cheating. Automated online proctoring is highly scalable and cost-effective when compared to using human proctors as it can operate without the involvement of human proctor [1]. Moreover, human proctors divided attention while proctoring a group of students can compromise the accuracy of proctoring.

Artificial Intelligence (AI) concerns with building smart machines capable of performing tasks that normally require human intelligence. AI can be used for many different things, detecting cheating can very well be one the task AI is set to take over. For an instance, computer vision is a field of artificial intelligence that defines the techniques to enable computers to interpret contents of images and videos to identify objects the way human eye would perceive. Computer vision technologies provide a viable way to monitor a large number of students at once through cameras instead of having human proctor.

Several research have investigated the behaviours of cheating students and how several movements can indicate cheating action. A study by [3] performed a comprehensive observation on test takers during exams. They have taken notice of test taker's head pose and time delay during examination and how these behaviors directly correlate to that of a cheating action. They were able to present an algorithm that is able to predict cheating with an average accuracy of 75.6% based on those factors. This proves that proper construction of image recognition algorithms can present a feasible solution for automating proctoring during online exams.

The market offers several systems for online proctoring such as the ProctorU (https://www.proctoru.com/), Proctorio (https://proctorio.com/), and ExamSoft (https://examsoft.com/). ProctorU is a secure, live, online proctoring software that enables exam takers to take their exams remotely. ProctorU provides various kinds of online proctoring services, including AI-based live proctoring. To take an AI invigilated exam, students must log into their account and install the ProctorU browser extension. At the time of exam, students will have to first go through an identity verification process. First, student will be prompted to enter a password to access the exam.

Next, ProctorU will ask students to align face with provided guidelines to take a photo. ProctorU will also ask to take a photo of student's ID card to verify their identity. Once verification is completed, students may officially begin with their exam. Next, during the AI invigilated exams, ProctorU utilizes AI to monitor exam sessions automatically, noting down any behavioural or technical anomalies during the exams while a human proctor validates those events. While ProctorU is robust in detecting cheating, the exam takers reportedly experience latency before starting a test. They also reported that ProctorU is too sensitive. Even littlest noises can flag the system. In addition, the verification process is lengthy, which can take up to 15 minutes.

Proctorio is one of the most prominent online proctoring software platforms that is used by many institutions. It detects suspicious activity through machine learning and advance facial detection technologies and flags these behaviours to the instructor. When sitting for a Proctorio exam, students only need to download the Proctorio extension in their browser. During the exam, Proctorio captures students' movements through their cameras and flag any suspicious activity. After the exam is submitted, the instructor can review the exam recordings and the flagged behaviours. Nonetheless, instructor may only review flagged events after exams have been submitted, not in real-time. The stored recordings of each student can also be costly and invasive.

ExamSoft is an assessment platform that allows education institutes and instructors to conduct secure assessment. In ExamSoft, students will have to download the exams 48 hours before the exam. When an exam is opened, all Internet access is blocked. Questions are automatically randomized to prevent exam takers from sharing answers by question. ExamSoft also provides AI re- mote proctoring solution in one of its suites of solutions, the ExamMonitor. ExamMonitor is an AI that reviews video and audio files recorded from an exam and detects for any anomalies. When unusual behaviour's are detected, it flags for further review. Similar to Proctorio, ExamSoft instructors may only review flagged events after exams have been submitted and it requires a professional proctor to review the AI flagged events.

3. Methodology

This paper proposed an automated online proctoring system called the ProctorEx. ProctorEx is developed based on the Extreme Programming (XP) soft- ware development methodology. Every development process in XP begins with the planning, followed by the iterations consisting of four primary phases: coding, testing, and listening. Figure 1 shows the phases in XP methodology.

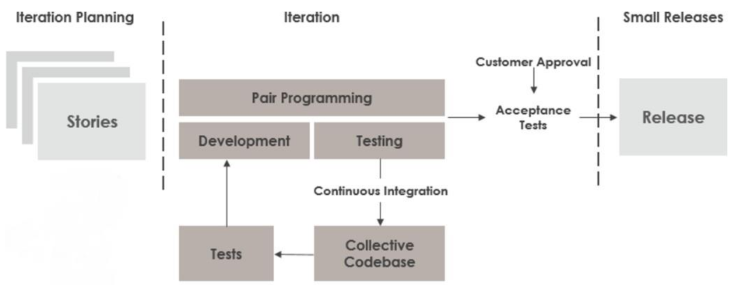

Fig. 1. Extreme Programming (XP) methodology (Extreme Programming (XP) vs Scrum, n.d.)

3.1 Iteration Planning Phase

The Iteration Planning phase corresponds to the analysis and design activities in developing ProctorEx. Figure 2 shows the Use Case Diagram for ProctorEx to illustrate the application functionalities and interactions.

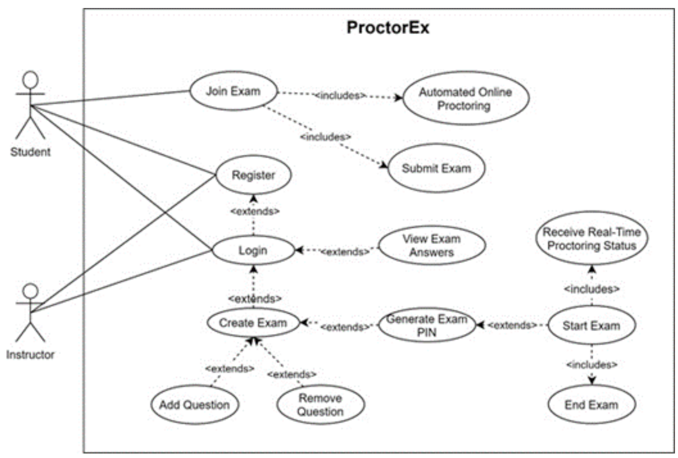

Fig. 2. Use case diagram for ProctorEx

Based on the Use Case Diagram, ProctorEx has four (4) main modules, which are face detection, gaze tracking, head pose estimation, and browser monitoring. The users are the student and the instructors. The students are able to create account, login and logout, join and take exam, and finally submit exam answers. Meanwhile, instructors are able to create and distribute exam, conduct online exam, oversee exam, receive real-time proctoring alert, and view exam answers in addition to create account as well as login and logout from the system.

3.2 Iteration Phase

In the Iteration phase, the coding and testing processes iterate. The core feature of ProctorEx is its fully automated proctoring functionality. During an exam in ProctorEx, the application will monitor students through their webcams and detect any suspicious movement or activity. The application leverages various methods and AI algorithms to perform the following actions automatically:

– Face Detection to detect when there is more than one person or when user moves away from computer (webcam).

– Gaze tracking to detect when user is looking away from screen.

– Head Pose Estimation to detect when user is facing away from screen.

– Browser activity monitoring to detect when user switches tab or splits screen.

Figure 3 shows the backend architecture for ProctorEx that highlights Face Detection, Gaze Tracking, and Head Pose Estimation.

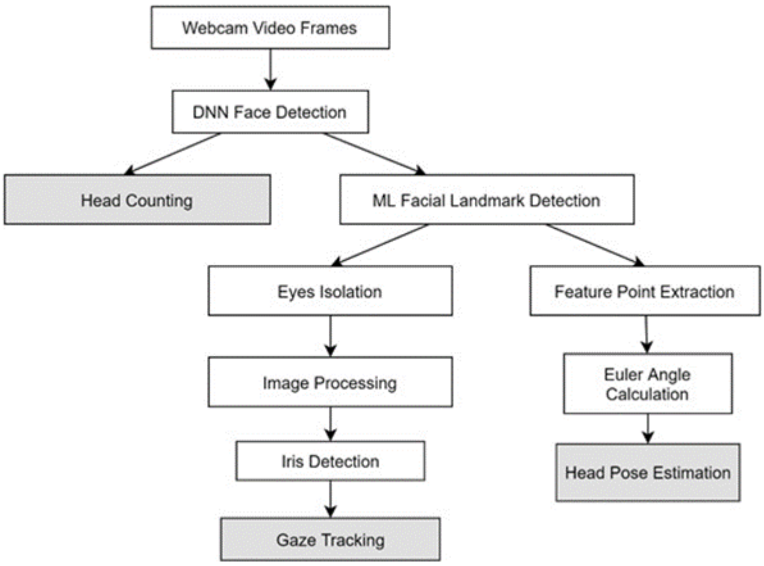

Fig. 3. Backend architecture for ProctorEx

Based on Figure 3, once the student's webcam video is acquired from frontend side, the frames are sent to the backend for face detection using deep neural network (DNN). Faces detected are counted and the facial landmark for the faces are extracted using Machine Learning (ML) algorithms. The landmarks are then used to perform image processing to detect the irises for gaze tracking and to calculate the position of the head. The following subsections will describe the details of the functionalities.

Face Detection this system receives webcam video and perform face detection. The first processing performed by the backend server is face detection. A thorough comparison was done by the researcher on various face detection algorithms before deciding to implement OpenCV Deep Neural Network (DNN) as it had good accuracy with fast execution time. For face detection, ProctorEx uses a pre-trained face detection model, the Caffe model. To use the model, the prototxt file which defines the model architecture and the caffe model file which contains the weights for DNN layers is required.

Face detection is performed by setting the blob through the net object. From there, a for loop is used to loop over the detected faces. The researcher has set the confidence threshold to 80 percent and filters out detection that has confidence score lower than the threshold. The face detector outputs the list of bounding boxes of all faces detected in the frame. Finally, the coordinates for the x, y, width and height of the bounding box are obtained.

Once face is detected, the facial landmark detected using Dlib. Dlib is an open-source C++ library that contains a variety of machine learning algorithms, including its pre-trained facial landmark detector. The facial landmark detector is able to accurately predict the location of 68 coordinates mapping to facial structures in real time. The 68 coordinates are visualized in Figure 4 as follows.

Fig. 4. Dlib facial landmarks (Pandey, 2021)

Gaze Tracking Once the facial landmarks have been obtained, the landmarks for the eye region can be used to isolate the eyes from the rest of the face. This is essentially the first step in gaze tracking. To perform this isolation, a mask is created using the coordinates of each eye obtained from the eye landmarks. The mask is then used to filter the eyes from the rest of the image. Next, the eye region is cropped from the image. The extreme points, which are the top left and bottom right coordinates of the eye, are used to extract the rectangular crop around the eyes.

The cropped image containing the eye is used to detect the iris. The iris is typically darker than the rest of the eye, therefore by doing binarization, the system should be able to extract the iris by removing the surroundings. To do that, the system must find a suitable threshold that would be used for the binarization process. If the threshold is too low, it might leave too many noises intact. If the threshold is too high, it will remove the iris region along with the noises. Once system has binarize the eye image with the best threshold, the iris should be white while the rest of the image is black. OpenCV findContours function is used to find white object, which should be the iris, from black back- ground. Finally, OpenCV moment’s function is used to find the coordinates for the centroid of the iris contour.

Due to varying reasons like changes of lightning or head posture, the accuracy of iris detection can be greatly affected. To improve the accuracy, whenever the iris is not detected, the system removes the first threshold from the list of 20 thresholds and recalculate a new threshold. This allows the system to continuously recalibrates itself to be able to detect irises under different conditions. Once ProctorEx has detected the iris, the next step would involve tracking the position of the iris along the eye. To do so, the system uses the x coordinate of the detected iris in the eye image and divides it with the width of eye image

This process returns a number between 0 and 1, with numbers leaning to- wards 0 indicating that iris is positioned towards right and numbers leaning to 1 being left. This value is used to track student’s gaze, determining whether student is looking left or right based on the position of their iris. To detect if students are looking down, the system calculates the vertical length at the center of the eye. As visualized in Figure 5, when eyes are closing or looking down, the vertical length decreases.

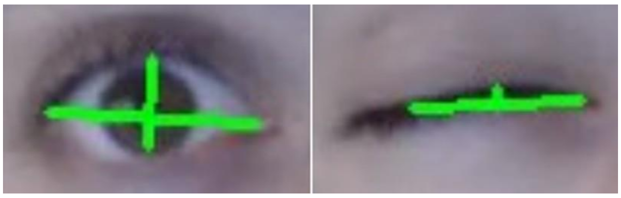

Fig. 5. Vertical length of eye (Canu, n.d.)

Note that the coordinates for the landmarks of the eyes are used to calculate the horizontal and vertical length of the eyes by calculating their distances.

For example, system finds the horizontal length of left eye by calculating distance between coordinates of landmarks 37 and 40. The vertical length is found by calculating the distance between the midpoint of landmark 38 and 39 and the midpoint of landmark 41 and 42. The ratio of horizontal and vertical length is used to determine if the eyes are closed. If the average ratio of left and right eye is larger than a specified threshold, system detects it as closing. If eyes are closed five consecutive frames, the system detects it as looking down.

Head Pose Estimation To estimate the head pose, five feature points are used: the corners of eyes, the tip of nose, the corners of mouth, and the chin. In ProctorEx, OpenCV’s solvePNP function is used to estimate the pose of 2D and 3D points with respect to the camera. The Euler angles are then extracted using the rotation vector and the translation vector values returned from the solve PNP function.

The Euler angles are three angles that describes the rotations around three independent axes. The angles, named “pitch”, “yaw”, and “roll” can be used to estimate the pose of the head. The pitch angle, which rotates around X axis, corresponds to face looking up and down. The Yaw angle, which rotates around Y axis, corresponds to face turning left and right. The roll angle, which rotates around Z axis, corresponds to face tilting left and right. These yaw angle from OpenCV decompose Project Matrix function is used to detect when student is facing left and right as shown in Figure 6.

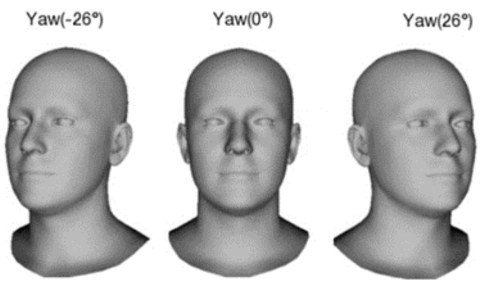

Fig. 6. Yaw angles (Kostyaev, 2020)

If yaw angle is lower or higher than the specified thresholds, system detects it as facing away from screen.

3.3 Small Releases Phase

In the Small Releases phase, ProctorEx is released as a prototype as further deliberated in the following section.

4. Prototype

ProctorEx is a fully automated online exam proctoring web application that enables users to conduct online exams in a convenient, secure, and cost-effective manner. The application enables instructors to easily create and distribute proctored exams and students to sit for those exams. To use ProctorEx, users can simply access the application on their web browsers. No installation is required for any part of the application. This section will present ProctorEx user interface for creating, starting, ending, and monitoring online examinations.

4.1 Create, Start, and End Exam

Figure 7 shows the main page of ProctorEx with navigation bar for “Create Exam”, “Join Exam”, “Login”, and “Sign up”. To create an exam using ProctorEx, instructors must create an account and login.

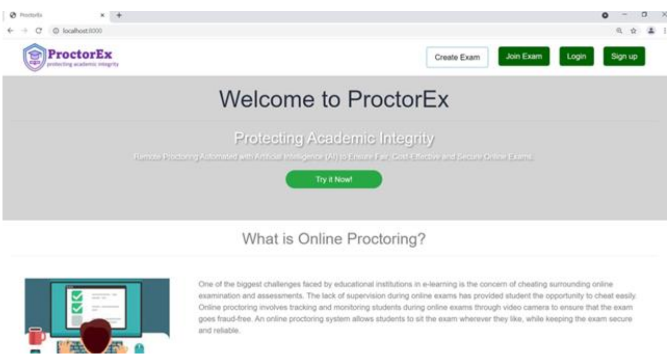

Fig. 7. ProctorEx landing page

Once the exam has been created via ProctorEx, the instructors may dis- tribute the exams to students through the exam PIN generated by ProctorEx. Once the students have joined the exam using the Exam PIN, instructor can click "Start Exam" button to officially start the exam. Once the instructor started the exam, the questions will immediately be displayed to students. When students submitted an exam, instructors may view their answers through ProctorEx. The exam ends either when student clicks the "submit" or when instructor clicks "End Exam". When exam ends, both users are redirected back to home page. To view answers submitted for an exam, instructor needs to click "Answers" in the side navigation menu to navigate to the answers page. The interface for these processes are shown in Figure 8.

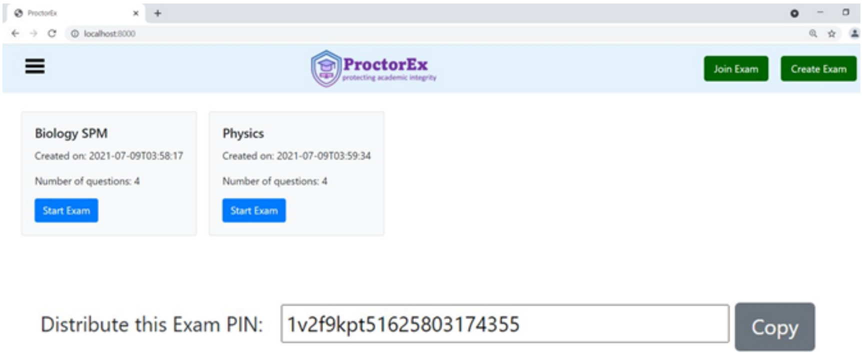

Fig. 8. Exam card and pin component

4.2 Monitor Exam Activities

The main advantage of ProctorEx is the capability for monitoring students behavior while taking exams real-time as shown in the Figure 9. During the exam, the system monitors students through their webcam and browser activity to detect possible cheating actions.

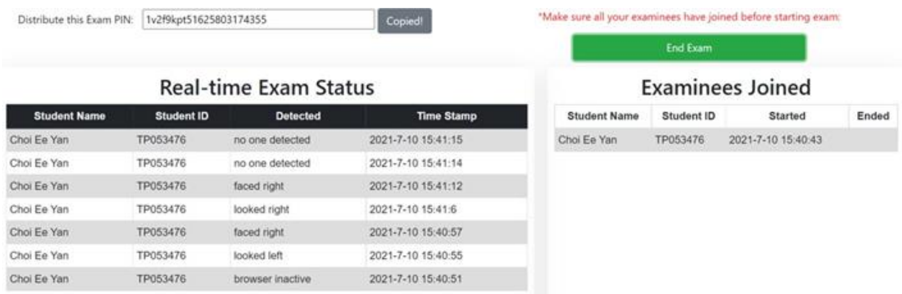

Fig. 9. Real-time automated proctoring status updates

Figure 9 shows the time stamp for each movement by the exam candidate in real-time. If suspicious activity is detected, the system sends an alert to the instructor. Any forms of detections are shown in the "Real-time Exam Status" table at instructor's exam page. There are 8 distinct types of proctor alerts that can be flagged to the instructors as the following. When any of the above activity was detected, the application immediately flags the activity to the instructor.

– Tab switching or split screen is detected

– Student's gaze is detected looking left

– Student's gaze is detected looking right
– Student's gaze is detected looking down
– Student's head is detected facing right
– Student's head is detected facing left
– No student is detected from webcam
– More than one person is detected from webcam

Figure 10 shows the demonstration process for face detection, facial landmark detection, gaze tracking, and pose estimation to users.

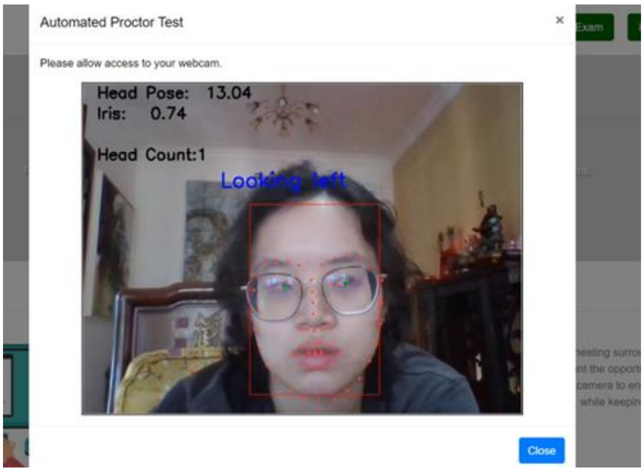

Fig. 10. ProctorEx on live test

5. Evaluation

As Extreme Programming methodology is incremental and work was carried out through subsequent iterations, different types of testing are done throughout system development life cycle, which are Unit Testing, Integration Testing, User Acceptance Testing, and System Testing. In order to evaluate ProctorEx, two types of tests were involved: Unit Testing and User Acceptance Testing (UAT). A Unit Testing is the software testing of individual units or modules of a software. This testing is crucial as it ensures that each unit is working accordingly before moving on to other types of testing. Doing frequent Unit Testing ensures that errors can be identified and fixed earlier in the development cycle which save cost and time. In this project, Unit Testing is executed during development of code by isolating sections of code in the program and tests for correctness.

A User Accepting Testing (UAT) is a software test carried out by end users to validate the usability of the system and to acquire feedback from users. Users are crucial to software development as they are the ones who would be using the software. Therefore, it is important to consider their thoughts and opinions and to make adjustments according to their preferences. In this project, UAT was carried out by selected people at the end of each iteration to receive feedbacks. Figure 11 shows one response from a user who tested ProctorEx.

User Acceptance Test (UAT) for ProctorEx						
Name: Tan Hong Peng Occupation: Teacher Date: 10/07/21 Start Time: 3:30 pm End Time: 4: 00 pm						
Legend: 0-Poor, 5-Excellent						
	0	1	2	3	4	5
17. Interface Design					X	
18. Ease of Use						X
19. Meeting Objectives						X
20. Free from Bugs						X
21. Input Validation						X
22. Performance (Speed and Loading time)				X		
23. Accuracy (Detection and Tracking)					X	
24. Overall Satisfaction					X	
Comments from Tester:	Speed subject to internet connection. Timing of detection was slightly affected due to slow internet speed at the time of testing.					
Action to be taken by developer:	The developer decreased the frame rate for processing to reduce amount of computation which increases the speed.					

Fig. 11. Sample response from one UAT conducted

In particular, three users were chosen to conduct the UAT. The first user who performed the user acceptance test had identified some errors that was later re- solved. Minor changes were also made to the frontend of the system based on first user's feedbacks to improve the system's overall usability. The second user who tested the system reveals that it can be slow when internet connection quality was below expectation. The solution to this feedback was to reduce webcam frame rates required for processing in order to lower down the computational effort and improve system's overall performance. The final tester has provided insightful feedbacks to be included in future enhancements.

I CONCLUSION

This paper described an automated online proctoring system called the ProtorEx, which capable for face detection, gaze tracking, head pose estimation, and browser monitoring. The main advantage of ProctorEx is that it relies on image processing technology, therefore enable the instructors to view the examination candidates real-time. It does not require third party intervention and is incredibly simple to use for both instructors and students. Instructors do not need to go through lengthy process to use ProctorEx services and students do not need to go through lengthy verification process before an online exam.

However, there are several challenges regarding the practicability of ProctorEx. First and foremost, the implementation of such system in any online exams must ensure that each student have a functional webcam since system relies on webcam video feed to perform automated proctoring. There are also concerns regarding students' privacy as they must provide the system access to their webcams during exams. Additionally, to ensure that the system will not falsely accuse students of cheating, good algorithms must be developed and implemented to ensure that detections and tracking is accurate. This is exceptionally important because students who have been monitored by similar systems have experienced constant cheating accusation which led to excessive stress during online exam [6]. Moreover, developing an accurate algorithm is especially challenging as different lightning condition and other environmental factors can greatly affect the accuracy of detection and tracking.

Nonetheless, ProctorEx is able to increase the prevalence of online exams by reducing the cost and time of staff allocation, venue preparation, travelling, paper printing and others associated with face-to-face exams. The system will also improve staff productivity by eliminating the need of many human proctors during online exams. Trusted online exams are expected to increase revenue of online educational institution as the system gives them the option of providing closed-book exams which can increase the credibility of these institutes. In the long run, ProctorEx is hoped to improve credibility of online exams and facilitate academic integrity.

ACKNOWLEDGMENT

This project is sponsored by Asia Pacific University of Technology & Innovation. Authors would like to thank explicit role in building the system.

REFERENCES

1. Atoum, Y., Chen, L., Liu, A.X., Hsu, S.D., Liu, X.: Automated online exam proc- toring. IEEE Transactions on Multimedia 19(7), 1609–1624, 2017.
2. Bilen, E., Matros, A.: Online cheating amid covid-19. Journal of Economic Behavior & Organization 182, 196–211, 2021.
3. Alaria, S. K., A. . Raj, V. Sharma, and V. Kumar. “Simulation and Analysis of Hand Gesture Recognition for Indian Sign Language Using CNN”. International Journal on Recent and Innovation Trends in Computing and Communication, vol. 10, no. 4, Apr. 2022, pp. 10-14, doi:10.17762/ijritcc.v10i4.5556.
4. Chuang, C.Y., Craig, S.D., Femiani, J.: Detecting probable cheating during online assessments based on time delay and head pose. Higher Education Research & Development 36(6), 1123–1137, 2017.
5. Kadhim, R. R., and M. Y. Kamil. “Evaluation of Machine Learning Models for Breast Cancer Diagnosis Via Histogram of Oriented Gradients Method and Histopathology Images”. International Journal on Recent and Innovation Trends in Computing and Communication, vol. 10, no. 4, Apr. 2022, pp. 36-42, doi:10.17762/ijritcc.v10i4.5532.
6. Danilyuk, J.: Academic Cheating Statistics: This Is What You Ought to Know, Sep 2019., https://unicheck.com/blog/academic-cheating-statistics
7. Dixon, L.: Does nontraditional, online education have a credibility problem?, Jul 2018., https://www.chieflearningofficer.com/2017/03/15/nontraditional-o nline-education-credibility-problem/

8. Harwell, D.: Cheating-detection companies made millions during the pandemic. now students are fighting back, 2020., https://www.washingtonpost.com/technology/2020/11/12/test-monitoring-student-revolt/
9. Hussein, M.J., Yusuf, J., Deb, A.S., Fong, L., Naidu, S.: An evaluation of online proctoring tools. Open Praxis 12(4), 509–525, 2020.
10. Labayen, M., Vea, R., Florez, J., Aginako, N., Sierra, B.: Online student authen- tication and proctoring system based on multimodal biometrics technology. IEEE Access 9, 72398–72411, 2021.
11. Chauhan, T., and S. Sonawane. "The Contemplation of Explainable Artificial Intelligence Techniques: Model Interpretation Using Explainable AI". International Journal on Recent and Innovation Trends in Computing and Communication, vol. 10, no. 4, Apr. 2022, pp. 65-71, doi:10.17762/ijritcc.v10i4.5538.
12. Song, L., Singleton, E.S., Hill, J.R., Koh, M.H.: Improving online learning: Stu- dent perceptions of useful and challenging characteristics. The internet and higher education 7(1), 59–70, 2004.
13. Kostyaev, D., 2020. Better rotation representations for accurate pose estimation. [online] Medium. Available at: https://towardsdatascience.com/better-rotation-representations-for-accurate-pose-estimation-e890a7e1317f
14. Visual-paradigm.com. n.d. Extreme Programming (XP) vs Scrum. [online] Available at: <https://www.visual-paradigm.com/scrum/extreme-programming-vs-scrum/>
15. Canu, S., n.d. Eye Blinking detection - Gaze controlled keyboard with Python and Opencv p.2 - Pysource. [online] Pysource. Available at: https://pysource.com/2019/01/10/eye-blinking-detection-gaze-controlled-keyboard-with-python-and-opencv-p-2/
16. Hoda, S. A. ., and D. A. C. . Mondal. "A Study of Data Security on E-Governance Using Steganographic Optimization Algorithms". International Journal on Recent and Innovation Trends in Computing and Communication, vol. 10, no. 5, May 2022, pp. 13-21, doi:10.17762/ijritcc.v10i5.5548.
17. Pandey, D., 2021. Eye Aspect Ratio (EAR) and Drowsiness detector using dlib. [online] Medium. Available at: https://medium.com/analytics-vidhya/eye-aspect-ratio-ear-and-drowsiness-detector-using-dlib-a0b2c292d706
18. J. . Hermina, N. S. . Karpagam, P. . Deepika, D. S. . Jeslet, and D. Komarasamy, "A Novel Approach to Detect Social Distancing Among People in College Campus", Int J Intell Syst Appl Eng, vol. 10, no. 2, pp. 153–158, May 2022.
19. M. . Parhi, A. . Roul, B. Ghosh, and A. Pati, "IOATS: an Intelligent Online Attendance Tracking System based on Facial Recognition and Edge Computing", Int J Intell Syst Appl Eng, vol. 10, no. 2, pp. 252–259, May 2022.
20. Yaghmaee F, Khammari R. Facial gender recognition، deferent approaches. sjis. 2020; 2 (3) :1-5
URL: http://sjis.srpub.org/article-5-60-fa.html